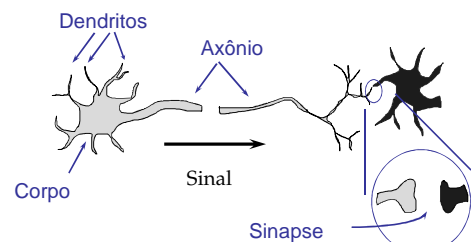
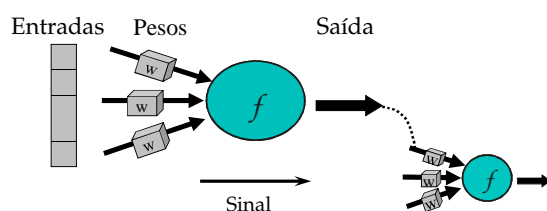
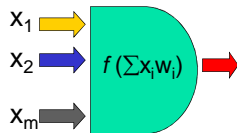
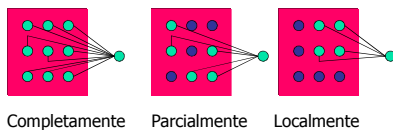
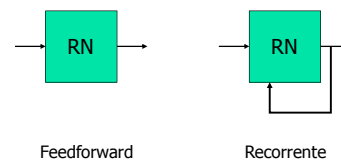
Mineração de Dados em Biologia Molecular
Redes Neurais Artificiais
André C. P. L. F. de Carvalho
Monitor: Valéria Carvalho

Principais tópicos
Introdução
Arquitetura
Aprendizado
Principais modelos
Perceptron
MLP
SVMs
André Ponce de Leon de Carvalho
2

Redes Neurais
Sistemas distribuídos inspirados no cérebro humano
Compostas por várias unidades de processamento ("neurônios")
Interligadas por um grande número de conexões ("sinapses")
Eficientes em várias aplicações
André Ponce de Leon de Carvalho
3

Neurônio Natural
Um neurônio simplificado:
Dendritos
Axônio
Corpo
Sinal
Sinapse
André Ponce de Leon de Carvalho
4

Neurônio artificial
Modelo de um neurônio abstrato
Entradas
Pesos
Saída
w
w
w
f
Sinal
w
w
w
f
André Ponce de Leon de Carvalho
5

Conceitos básicos
Principais aspectos das RNA
Arquitetura
Unidades de processamento (neurônios)
Conexões
Topologia
Aprendizado
Algoritmos
Paradigmas
André Ponce de Leon de Carvalho
6

Unidades de processamento
Funcionamento
Recebe entradas de conjunto de unidades A
Aplica função sobre entradas
Envia resultado para conjunto de unidades B
Entrada total
$u = \sum_{i=1}^{m} x_i w_i$
x_1
x_2
x_m
$f(\sum x_i w_i)$
André Ponce de Leon de Carvalho
7

Conexões
Definem como neurônios estão interligados
Codificam conhecimento da rede
Tipos de conexões:
Excitatória: ($w_{ik}(t) > 0$)
Inibitória: ($w_{ik}(t) < 0$)
André Ponce de Leon de Carvalho
8

Topologia
Número de camadas
Uma camada (Ex Perceptron, Adaline)
Multi-camadas (Ex MLP, RBF)
Completamente
Parcialmente
Localmente
André Ponce de Leon de Carvalho
9

Topologia
Arranjo das conexões
RN
RN
Feedforward
Recorrente
André Ponce de Leon de Carvalho
10

Aprendizado
Algoritmos de aprendizado
Conjunto de regras bem definidas para ensinar a rede a resolver um dado problema
Principais grupos
Correção de erro
Hebbiano
Competitivo
Termodinâmico (Boltzmann)
André Ponce de Leon de Carvalho
11

Aprendizado
Paradigmas de aprendizado
Como a RNA se relaciona com o ambiente externo
Principais tipos
Supervisionado
Não supervisionado
Reforço
Híbrido
André Ponce de Leon de Carvalho
12

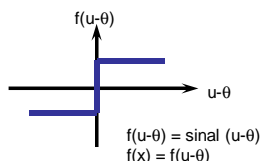
Tipos de atributos

- Redes Neurais trabalham apenas com números
  - Não aceitam dados categóricos
    - Precisam ser convertidos
  - Dificuldade para lidar diretamente com imagens
    - Precisa ser pré-processadas

Perceptron

- Desenvolvida por Rosemblat, 1958
- Utiliza modelo de neurônio de McCulloch-Pitts
  - Formularam matematicamente neurônios naturais
- Rede mais simples para classificação de padrões linearmente separáveis

Perceptron

- Treinamento
  - Supervisionado
  - Correção de erro
    - $w_i(t) = w_i(t-1) + \Delta w_i$
      - $\Delta w_i = \eta x_i(y - f(x))$
  - Induz hipótese ou função f(x)
- Rosemblat provou teorema de convergência
  - Se é possível induzir um classificador um conjunto de entradas, uma rede Perceptron induzirá

Perceptron

- Resposta / saída da rede
  - Aplica função limiar sobre soma total de entrada recebida por um neurônio

$$u = \sum_{i=1}^{m} x_i w_i$$

$$f(u) = \begin{cases} +1 & \text{if } u \geq \theta \\ -1 & \text{if } u < \theta \end{cases}$$

f(u-θ)

u-θ

f(u-θ) = sinal (u-θ)
f(x) = f(u-θ)

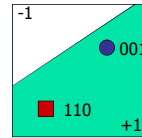
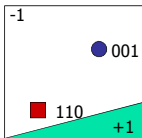
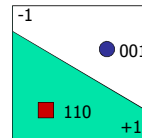
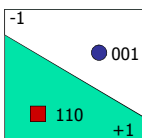
Algoritmo de treinamento

1 Iniciar todas as conexões com $w_i = 0$
2 Repita
Para cada par de treinamento (X, y)
Calcular a saída f(x)
Se ($y \neq f(x)$)
Então
Atualizar pesos do neurônio
Até o erro ser aceitável

Exemplo

- Dada uma rede Perceptron com:
  - Três entradas, pesos iniciais $w_1 = 0.4$, $w_2 = -0.6$ e $w_3 = 0.6$, e limiar (viés) $\theta = 0.5$:
    - Ensinar a rede com os exemplos (001, -1) e (110, +1)
      - Utilizar taxa de aprendizado $\eta = 0.4$
    - Definir a classe dos exemplos: 111, 000, 100 e 011

Exemplo
x1
x2
x3
w1
w2
w3
f(x)
Viés
Situação desejada
-1
001
110
+1
André Ponce de Leon de Carvalho
19

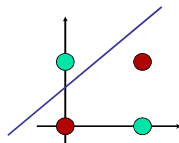
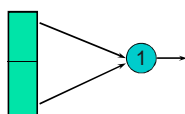
Exemplo - treinamento
a) Treinar a rede
a.1) Para o exemplo 001 (y = -1)
Passo 1: definir a saída da rede
u-θ = -1(0.5) + 0(0.4) + 0(-0.6) + 1(0.6) = 0.1
f(x) = +1 (uma vez 0.1 ≥ 0)
Passo 2: atualizar pesos (y ≠ h(x))
w0 = 0.5 + 0.4(-1)(-1 - (+1)) = 1.3
w1 = 0.4 + 0.4(0)(-1 - (+1)) = 0.4
w2 = -0.6 + 0.4(0)(-1 - (+1)) = -0.6
w3 = 0.6 + 0.4(1)(-1 - (+1)) = -0.2
-1
001
110
+1
André Ponce de Leon de Carvalho
20

Exemplo - treinamento
a) Treinar a rede
a.2) Para o exemplo 110 (y = +1)
Passo 1: definir a saída da rede
u-θ = -1(1.3) + 1(0.4) + 1(-0.6) + 0(-0.2) = -1.5
f(x) = -1 (uma vez -1.5 < 0)
Passo 2: atualizar pesos (y ≠ f(x))
w0 = 1.3 + 0.4(-1)(1 - (-1)) = 0.5
w1 = 0.4 + 0.4(1)(1 - (-1)) = 1.2
w2 = -0.6 + 0.4(1)(1 - (-1)) = 0.2
w3 = -0.2 + 0.4(0)(1 - (-1)) = -0.2
-1
001
110
+1
André Ponce de Leon de Carvalho
21

Exemplo - treinamento
a) Treinar a rede
a.3) Para o exemplo 001 (y = -1)
Passo 1: definir a saída da rede
u -θ = -1(0.5) + 0(1.2) + 0(0.2) + 1(-0.2) = -0.7
f(x) = -1 (uma vez -0.7 < 0)
Passo 2: atualizar pesos (y = f(x))
Como y = f(x), os pesos não precisam ser modificados
-1
001
110
+1
André Ponce de Leon de Carvalho
22

Exemplo - treinamento
a) Treinar a rede
a.4) Para o exemplo 110 (y = +1)
Passo 1: definir a saída da rede
u-θ = -1(0.5) + 1(1.2) + 1(0.2) + 0(-0.2) = +0.7
f(x) = +1 (uma vez 0.7 > 0)
Passo 2: atualizar pesos (y = f(x))
Como y = f(x), os pesos não precisam ser modificados
-1
001
110
+1
André Ponce de Leon de Carvalho
23

Exemplo - teste
Utilizar a rede treinada para classificar os exemplos 111, 000, 100 e 011
André Ponce de Leon de Carvalho
24

Exemplo - teste
b) Testar a rede
b.1) Para o exemplo 111
u-θ = -1(0.5) + 1(1.2) + 1(0.2) + 1(-0.2) = 0.7
f(x) = 1 (porque 0.7 ≥ 0)) ⇒ classe +1
b.2) Para o exemplo 000
u-θ = -1(0.5) + 0(1.2) + 0(0.2) + 0(-0.2) = -0.5
f(x) = -1 (porque -0.5 < 0) ⇒ classe -1
André Ponce de Leon de Carvalho
25

Exemplo - teste
b) Testar a rede
b.3) Para o exemplo 100
u-θ = -1(0.5) +1(1.2) + 0(0.2) + 0(-0.2) = 0.7
f(x) = 1 (porque 0.7 ≥ 0) ⇒ classe +1
b.4) Para o exemplo 011
u-θ = -1(0.5) + 0(1.2) + 1(0.2) + 1(-0.2) = -0.5
f(x) = -1 (porque -0.5 < 0) ⇒ classe -1
André Ponce de Leon de Carvalho
26

Exercício
Seja o seguinte cadastro de pacientes:
Nome Febre Enjôo Manchas Dores Diagnóstico
João sim sim pequenas sim doente
Pedro não não grandes não saudável
Maria não sim pequenas não saudável
José sim sim grandes sim doente
Ana sim não pequenas sim saudável
Leila não não grandes sim doente
André Ponce de Leon de Carvalho
27

Exercício
Ensinar uma rede do tipo Perceptron a distinguir:
Pacientes potencialmente saudáveis
Pacientes potencialmente doentes
Testar a rede para novos casos
(Luis, não, não, pequenas, sim)
(Laura, sim, sim, grandes, sim)
André Ponce de Leon de Carvalho
28

Problemas com Perceptron
0, 0 → 0
0, 1 → 1
1, 0 → 1
1, 1 → 0
1
André Ponce de Leon de Carvalho
29

Rede Multi-Layer Perceptron
Arquitetura de RNA mais utilizada
Uma ou mais camadas intermediárias de neurônios
Funcionalidade
Uma camada intermediária: qualquer função contínua ou Booleana
Duas camadas intermediárias: qualquer função
Originalmente treinada com o algoritmo Backpropagation
André Ponce de Leon de Carvalho
30

Backpropagation

- Procura reduzir os erros cometidos pela rede
  - Utiliza erro para ajustar valor dos pesos
- Erro de cada neurônio
  - Camada de saída
    - Saída desejada - saída produzida
  - Camadas intermediárias ???

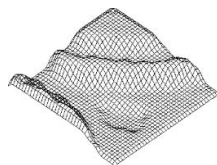
André Ponce de Leon de Carvalho 33

Backpropagation

- Procura reduzir os erros cometidos pela rede
  - Utiliza erro para ajustar valor dos pesos
- Erro de cada neurônio
  - Camada de saída
    - Saída desejada - saída produzida
  - Camadas intermediárias
    - Proporcional aos erros dos neurônios da camada seguinte conectados a ele

André Ponce de Leon de Carvalho 34

Backpropagation

- Treinamento
  - Supervisionado
  - Procura na superfície de erro onde o valor do erro é mínimo
    - Gradiente

André Ponce de Leon de Carvalho 35

Backpropagation

- Ajuste dos pesos
  - $\Delta w_{ij} = \eta x_i \delta_j$

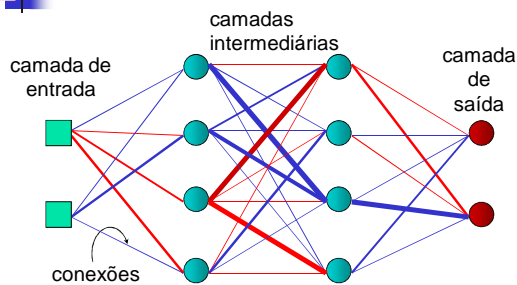
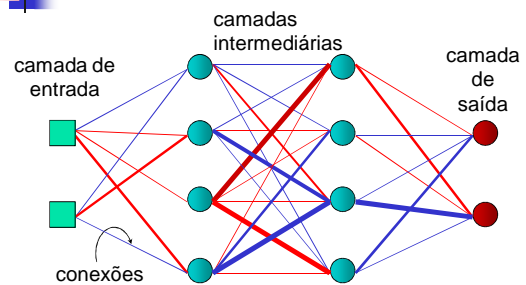
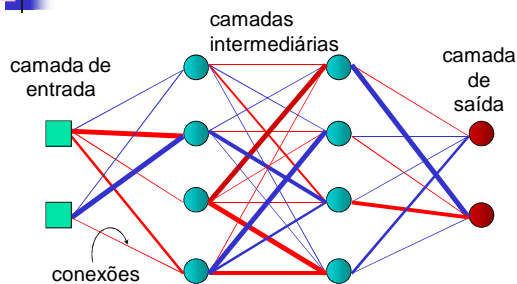
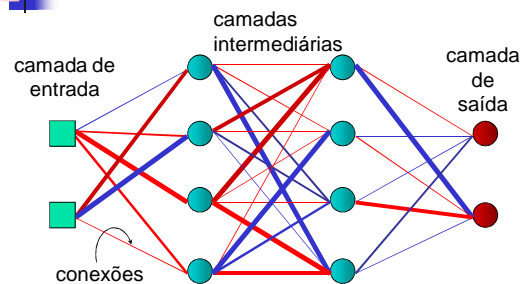
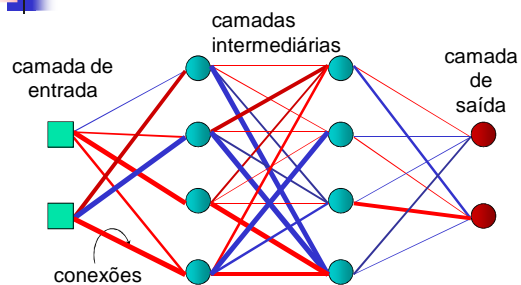
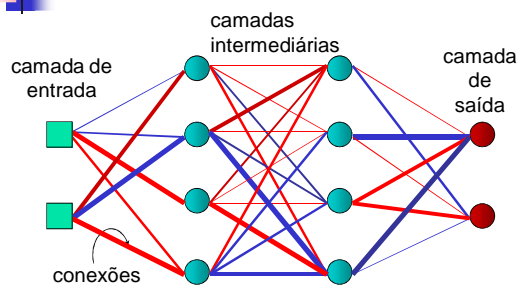
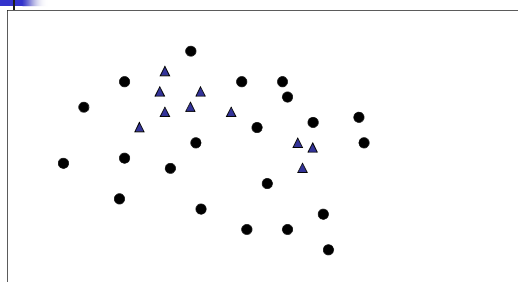
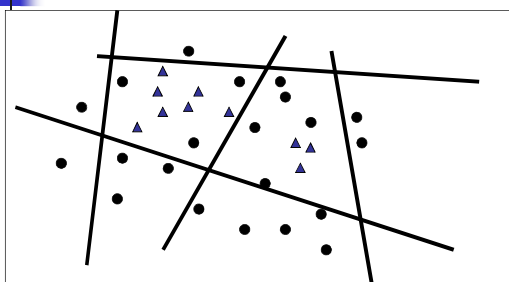
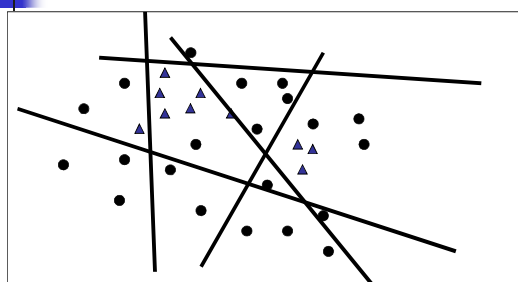
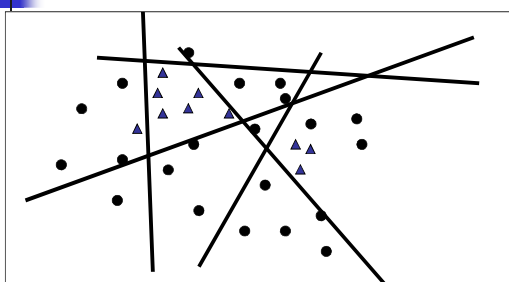
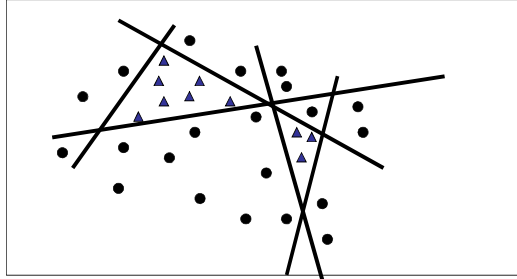
$$\delta_j = \begin{cases} f' erro_j & \text{se j for camada de saída} \\ f' \sum w_{jk} \delta_k & \text{se j for camada intemediária} \end{cases}$$

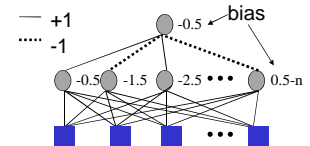
$$erro_j = \frac{1}{2} \sum_{q=1}^{c} (y_q - f(x)_q)$$

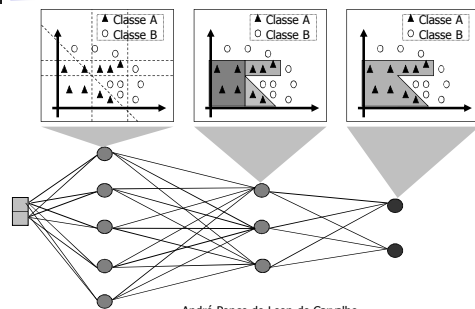
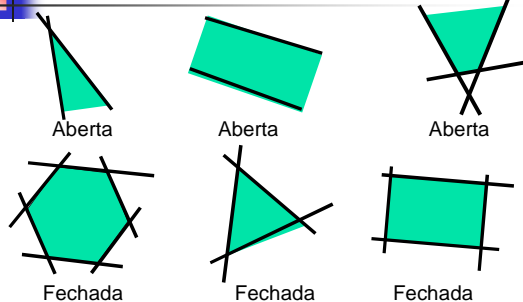
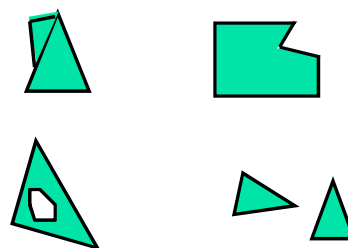
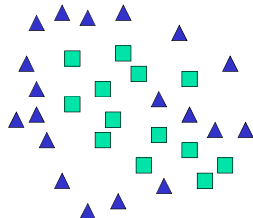
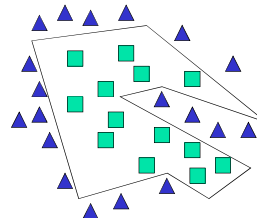
  - Se f for uma função sigmoidal, $f'(x) = f(x)(1-f(x))$
  - Treinamento não é garantido de convergir

André Ponce de Leon de Carvalho 36

Backpropagation
Função de ativação
Não linear
Diferenciável , contínua e, geralmente, não decrescente
Sigmoidal
$f(x) = 1/(1 + e^{-u_j(t)})$ (sigmoid logística)
$f(x) = \frac{(1 - e^{-u_j(t)})}{(1 + e^{-u_j(t)})}$ (tangente hiperbólica)
André Ponce de Leon de Carvalho
37

Treinamento
Iniciar todas as conexões com valores aleatórios
Repita
erro = 0
Para cada par de treinamento (X, y)
Para cada camada k := 1 a N
Para cada neurônio j := 1 a M_k
Calcular a saída $f_{kj}(x)$
Se k= N
Calcular soma dos erros de seus neurônios
Se erro > ε
Para cada camada k := N a 1
Para cada neurônio j:= 1 a M_k
Atualizar pesos
Até erro < ε (ou número máximo de ciclos)
André Ponce de Leon de Carvalho
38

Treinamento modificando pesos
camadas intermediárias
camada de entrada
camada de saída
conexões
André Ponce de Leon de Carvalho
39

Treinamento modificando pesos
camadas intermediárias
camada de entrada
camada de saída
conexões
André Ponce de Leon de Carvalho
40

Treinamento modificando pesos
camadas intermediárias
camada de entrada
camada de saída
conexões
André Ponce de Leon de Carvalho
41

Treinamento modificando pesos
camadas intermediárias
camada de entrada
camada de saída
conexões
André Ponce de Leon de Carvalho
42

Treinamento modificando pesos
camadas intermediárias
camada de entrada
camada de saída
conexões
André Ponce de Leon de Carvalho
43

Treinamento modificando pesos
camadas intermediárias
camada de entrada
camada de saída
conexões
André Ponce de Leon de Carvalho
44

Treinamento modificando fronteiras
André Ponce de Leon de Carvalho
45

Treinamento modificando fronteiras
André Ponce de Leon de Carvalho
46

Treinamento modificando fronteiras
André Ponce de Leon de Carvalho
47

Treinamento modificando fronteiras
André Ponce de Leon de Carvalho
48

Treinamento modificando fronteiras
André Ponce de Leon de Carvalho
49

Exercício
Dada a rede abaixo, que recebe como entrada um vetor binário de n bits e gera como saída um valor binário:
a) Indicar a função implementada pela rede abaixo:
b) Explicar papel de cada neurônio no processamento da função
Considerar função de ativação limiar (threshold) entrada/saída binária
+1
-1
bias
-0.5
-0.5
-1.5
-2.5
0.5-n
André Ponce de Leon de Carvalho
50

Exercício
Paridade
Uma das limitações do Perceptron levantadas por Minsky e Papert
Problema difícil
Padrões mais semelhantes requerem respostas diferentes
Usa n unidades intermediárias para detectar paridade em vetores com n bits
André Ponce de Leon de Carvalho
51

MLPs como classificadores
Classe A
Classe B
André Ponce de Leon de Carvalho
52

Regiões convexas
Aberta
Aberta
Aberta
Fechada
Fechada
Fechada
André Ponce de Leon de Carvalho
53

Combinações de regiões convexas
André Ponce de Leon de Carvalho
54

Combinações de regiões convexas
Encontrar fronteiras de decisão que separem os dados abaixo:
André Ponce de Leon de Carvalho
55

Combinações de regiões convexas
Encontrar fronteiras de decisão que separem os dados abaixo:
André Ponce de Leon de Carvalho
56

Exercício
Quantas camadas e pelo menos quantos nodos em cada camada possui a rede que divide o espaço de entradas das formas abaixo:
classe 1
classe 2
classe 1
classe 2
André Ponce de Leon de Carvalho
57

Unidades intermediárias
Número de camadas intermediárias necessárias
Funcionalidade desejada
Número de neurônios por camada
Distribuição dos dados
Early stop
Poda
André Ponce de Leon de Carvalho
58

Ajuste dos pesos
Por exemplo (online)
Por ciclo (batch)
Após apresentação de todos os exemplos de treinamento (ciclo)
Melhor alternativa depende da aplicação
Weight decay
André Ponce de Leon de Carvalho
59

Variações do backpropagation
Momentum
Quickprop
Newton
Levenberg Marquardt
Super Self-Adjusting Backpropagation (superSAB)
Métodos de gradiente conjugado
André Ponce de Leon de Carvalho
60

Backpropagation Momentum

- Treinamento
  - Supervisionado
  - $w_{ij}(t)= w_{ij}(t-1) + \Delta w_{ij}$ + momentum
    - $\Delta w_{ij} = \eta x_i \delta_j$
    - Momentum = $\alpha(w_{ij}(t-1) - w_{ij}(t-2))$
      - Aceleração

Deep Networks

- Redes neurais em geral têm 1 ou 2 camadas intermediárias
  - Mais camadas levam a soluções pobres
- Complexidade em teoria de circuitos
  - Sugere que arquiteturas produndas podem ser muito mais eficientes
    - Quando tarefa é complexa e existem dados suficientes para capturar essa complexidade
    - Necessidade de algoritmos apropriados

Deep Networks

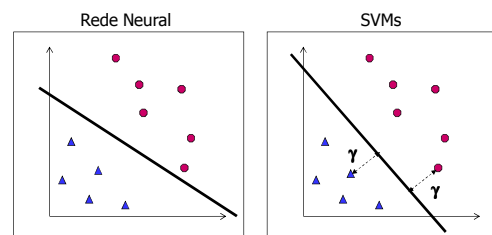
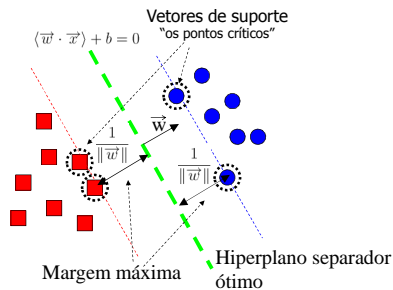
- Abordagens para treinamento
  - Adicionar camadas construtivamente
    - Cada camada transforma entrada da camada anterior
    - Torna tarefa de aprendizado cada vez mais fácil
  - Utilizar aprendizado não suprevisionado para cada camada
  - Treinar a rede toda de uma vez

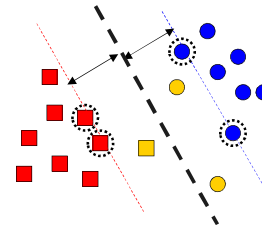
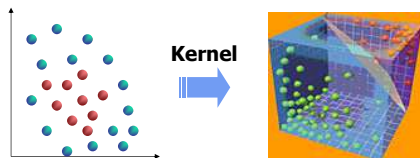
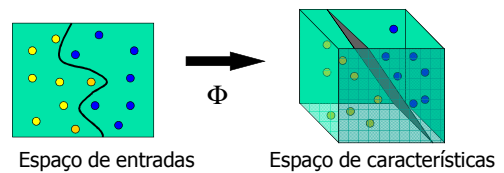
Outras redes

- Adaline
- RBF
- SOM
- GNG
- ART
- TDNN
- SVM ?

Máquinas de Vetores de Suporte (SVMs)

- Baseadas na Teoria do Aprendizado Estatístico
  - Vapnik e Chervonenkis em 1968
- Estratégia básica
  - Encontrar um hiperplano que maximize margem de separação (margem larga)
    - Distância a um conjunto de "vetores de suporte"
    - Reduz erro de generalização
      - Minimização do risco estrutural

Máquinas de Vetores de Suporte (SVMs)

Máquinas de Vetores de Suporte (SVMs)
Vetores de suporte
"os pontos críticos"
$\langle \vec{w} \cdot \vec{x} \rangle + b = 0$
$\vec{w}$
$\frac{1}{\|\vec{w}\|}$
$\frac{1}{\|\vec{w}\|}$
Hiperplano separador ótimo
Margem máxima
11/10/2012
André de Carvalho - ICMC/USP
67

Variáveis de folga
Slack variables
André Ponce de Leon de Carvalho
68

Máquinas de Vetores de Suporte (SVMs)
Problemas lineares
Generalização para problemas não lineares
Mapeamento de dados de entrada para um espaço de maior dimensão utilizando funções kernel
Kernel
11/10/2012
André de Carvalho - ICMC/USP
69

Fronteiras mais complexas
Φ
Espaço de entradas
Espaço de características
André Ponce de Leon de Carvalho
70

Conclusão
Redes Neurais
Sistema nervoso
Muito utilizadas em problemas reais
Várias arquiteturas e algoritmos
Magia negra
Caixa preta
André Ponce de Leon de Carvalho
71

Perguntas
André Ponce de Leon de Carvalho
72